

# HTML / CSS / JavaScript

Thomas FORGIONE  
`thomas.forgione@enseeiht.fr`

septembre 2017

# Les TPs

Les TPs

<https://tforgione.fr/javascript/>

## 1 HTML5

## 2 CSS

## 3 JavaScript

- Introduction
- Le langage
- Le DOM
- Canvas

# Historique

- 1991 Première version d'HTML (Tim Berners-Lee)
- 1994 HTML2 standard de W3C : *World Wide Web Consortium* :  
<http://www.w3.org/>
- 1996 HTML3 permettant de gérer tables, applets, scripts
- 1998 HTML4 permettant l'intégration de CSS *Cascading Style Sheets*
- 1998 W3C abandonne HTML : XHTML
- 2003 Mais **WHATWG** *Web hypertext App. Technology Working Group*  
Groupe de travail sur HTML5
- 2011 W3C accueille WHATWG  
Conclusion : 2 standards par W3C et par WHATWG
- 2014 HTML5

# Hypertext Markup Language

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre de la page</title>
  </head>
  <body>
    <h1>Mon premier titre</h1>
    <p>Mon premier paragraphe</p>
  </body>
</html>
```

## Mon premier titre

Mon premier paragraphe

# Les balises

- `<balise>...</balise>`
- Balises existantes :
  - Éléments de contenu : `<h1>...<h6>`, `<a>`, `<em>`, `<strong>`, `<mark>`, `<p>`, `<pre>`, `<ol>`, `<ul>`, `<table>`
  - Éléments de formulaires : `<form>`, `<input>`, `<textarea>`, `<select>`, `<option>`, ...
  - Éléments génériques : `<div>`, `<span>`
  - Éléments sémantiques : `<header>`, `<nav>`, `<footer>`, `<section>`, `<article>`, `<aside>`
  - Éléments multimédia : `<img>`, `<audio>`, `<video>`

# Les attributs

- `<balise attribut="valeur" autre-attribut>`
- Exemple : une vidéo

```
<video width="320" height="240" controls>  
  <source src="video.mp4" type="video/mp4">  
  <source src="video.ogv" type="video/ogg">  
  Votre navigateur ne supporte pas la balise video  
</video>
```

- Exemple : un lien  
`<a href="/toto.html">Cliquez ici !</a>`
- Attributs importants : `id` et `class`

# Hello world

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre de la page</title>
    <link rel="stylesheet" href="mon-fichier.css">
  </head>
  <body>
    <!-- Mon commentaire -->
    <h1>Mon premier titre</h1>
    <p>Mon premier paragraphe</p>
    <script src="mon-script.js"></script>
  </body>
</html>
```



# Cascading Style Sheets

- Ça sert à faire joli
- le HTML pour le contenu (avec du sens)
- le CSS pour la forme (la façon dont le contenu va s'afficher)

# Hello world avec CSS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre de la page</title>
    <link rel="stylesheet"
          href="hello-world.css">
  </head>
  <body>
    <h1>Ceci est un titre</h1>
    <p>Ceci est un paragraphe</p>
  </body>
</html>
```

```
body {
  background-color: powderblue;
}
h1 {
  color: red;
}
p {
  color: blue;
}
```

**Ceci est un titre**

Ceci est un paragraphe

# Explications

```
selecteur {  
    propriete1: valeur;  
    propriete2: valeur;  
}  
  
// Exemples  
#monElement {  
    // S'applique à l'élément HTML d'id "monElement"  
}  
  
.mesElements {  
    // S'applique aux éléments HTML de class "mesElements"  
}
```

# Javascript ?

## C'est quoi ?

- C'est pas Java
- Langage de programmation côté client (*quoique...*)
- Typage faible
- Typage dynamique

## Mais pourquoi ?

- À la base, éviter de nombreuses requêtes vers le serveur pour rien
- Pour faire des effets stylés
- Pour avoir une application simple côté client (convertisseurs, calettes, etc...)
- Pour les interfaces multimédia (video, mais aussi 3D avec WebGL)
- Pour tout et n'importe quoi depuis `nodejs`

# L'évolution de JavaScript

**ES1/ES2** Les débuts de JavaScript (1997,1998)

**ES3** Meilleures regex, strings, apparition du try / catch (1999)

**ES4** Abandonné à cause de modifications qui n'ont pas été appréciées

**ES5** Clarifie beaucoup d'ambigüités de la 3<sup>ème</sup> version (2009), version la plus répandue

**ES6** Beaucoup de nouvelles choses (2015, pas encore parfaitement supporté partout)

# Déclarations

## Déclaration de variables

`var` ou `let` (ES6)

## Les types

- non défini, `undefined`, `null`
- Nombres `var toto = 42; let pi = 3.14;`
- Booleéns `var monBooleen = true;`
- Chaines de caractères `var name = "Hello world !";`
- Tableaux `var array = [1,2,"Toto"];`
- Objets `var monObjet = {x:0, y:0, nom:"Toto"};`

**Une variable non déclarée sera définie comme globale**

# let vs var

```
var i = 0;

if (true) {
  var i = 1;
  // i vaut 1
}

// i vaut 1
```

var scope au niveau des fonctions

```
let i = 0;

if (true) {
  let i = 1;
  // i vaut 1
}

// i vaut 0
```

let (ES6) scope au niveau des blocs

# Les fonctions

Deux façons de définir une fonction :

```
function maFonction(monParametre1, monParametre2) {  
    // Des trucs  
    return monRetour  
}
```

```
var maFonction = function(monParametre1, monParametre2) {  
    // Des trucs  
    return monRetour  
}
```

**Attention à l'initialisation :**

- Dans le premier cas, la fonction sera disponible partout
- Dans le deuxième cas, elle sera disponible seulement après la ligne où l'affectation est réalisée



# Les fonctions existantes

- `alert` : affiche un message dans une boîte de dialogue
- `console.log` : affiche un message dans la console (F12)
- `prompt` : affiche une question dans une boîte de dialogue

# Les spécificités de JavaScript

- Les comparaisons

```
1 == "1" // true
1 === "1" // false

null == undefined // true
null === undefined // false

[1,2] == [1,2] // false
```

- Les foreach : array étant un tableau

```
for (var i = 0; i < array.length; i++) {
    var elt = array[i];
}

for (var key in array) {
    var elt = array[key];
}

for (var elt of array) { // ES6
}
```

# Les bizarreries de JavaScript

- Non défini, undefined, null

```
toto === undefined // ReferenceError: toto is not defined
typeof toto === "undefined"; // true

var toto; // undefined

toto == undefined;
toto === void 0;
```

- Juste pour rire  
Démonstration live

# Document Object Model

- Interface qui permet de manipuler les documents HTML
- Les éléments du DOM sont des objets

# Les éléments du DOM

## window

- Objet global qui représente la fenêtre
- Contient toutes les variables globales

## document

- Représente la racine de la page web
- Il permet d'accéder à tous les éléments HTML
- `window.document === document`

# Pour être plus concrêt

```
document.getElementById(id)
```

Renvoie un objet de type `Element` qui contient des propriétés et méthodes pour faire des trucs stylés

```
document.getElementsByClassName(class)
```

Renvoie un tableau d'`Elements` qui correspondent à la classe

# Les Elements

```
<!doctype html>
<html>
  <head>
    <title>Ma page</title>
  </head>
  <body>
    <p id="paragraphe"></p>
    <script src="script.js"></script>
  </body>
</html>
```

```
var p = document.getElementById('paragraphe')
p.innerHTML =
  'Ceci est <strong>mon</strong> paragraphe !';
```

Ceci est **mon** paragraphe !

Exemple avec `element.innerHTML`

# Les évènements

## Event

- Représente un évènement
- Possibilité d'appeler automatiquement une fonction sur un event

```
element.addEventListener('click', function()... );
```

Exemple : calculator.html



# Canvas

## La balise <canvas>

- Permet de dessiner des éléments
- Coin supérieur gauche : (0,0)

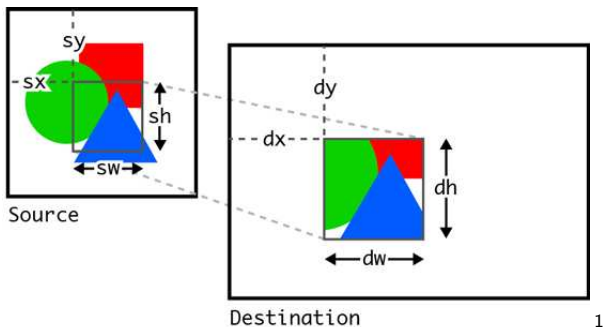
## Le context

- S'obtient par `canvas.getContext('2d')`
- Peut aussi être un `WebGLRenderingContext` (`canvas.getContext('webgl')`)
- Contient plein de fonctions et d'attributs pour dessiner

Exemples : `canvas-example.html`, `mouse-move.html`

# drawImage

```
ctx.drawImage(img, sx, sy, sw, sh, dx, dy, dw, dh);
```



Exemple : [lena.html](#)

1. [code.tutsplus.com/tutorials/canvas-from-scratch-pixel-manipulation--net-20573](http://code.tutsplus.com/tutorials/canvas-from-scratch-pixel-manipulation--net-20573)

## Encore quelques fonctions

```
setTimeout(function() {  
    console.log("Hello world \!");  
}, 1000);
```

```
function render() {  
    requestAnimationFrame(render);  
  
    // Do stuff  
}
```

Exemples : `requestAnimationFrame*.html`

# Les TPs

`https://tforgione.fr/javascript/`